第 12 部

音声

第 1 章

序論

音は昔から、人間と深い関わりあいを持っている。人間にとって有用な音もあれば、騒音と呼ばれるようなものもある。それではもし、この世から音を無くしてしまったらどうなるだろう。おそらく我々は、大切な情報伝達手段の１つを奪われて大変な不自由を強いられることになる。また音の中でも特に人の声、つまり音声は人間にとって最も手軽で重要なコミュニケーションの手段である。昔の漫画をみているとロボットが人間の言葉を話す場面が出てきた。ひと昔前まではそれは単なる夢であった。しかし、現在ではそれも夢ではなくなりつつある。その証拠に、最近いろいろな所で音声利用が始められている。例えば、電卓であるとか、目覚し時計であるとか。最近では炊飯器にさえ利用されるほどである。何故これほど音声が利用されるようになったのか。勿論、半導体技術の発達といった、技術的な面からの理由もある。しかしそれ以上に、音声が人間にとって大切な情報伝達の手段であるということが、大きな要因になってはいないだろうか。

とはいうものの、それは始められたばかりであり周りにはまだまだ利用可能な場所が数多く残っている。コンピュータもそのうちの一つである。しかし、コンピュータに音声を導入しようという試みは既に、様々なところで始められている。したがって、ただ単に音声をコンピュータに導入しようというのでは、何ら新しいことをすることにはならない。ところで、近年コンピュータネットワークなるものが世の中に出回り始めた。それが世に出回ってからまだそれほど年月はたっていない。ところが現在では、それは個人にとって、また特に企業にとって、欠かすことのできない重要な情報収集の手段となっている。それは、今後も２１世紀の高度情報化社会に向けて更に発展するであろうと思われる。ならば、このコンピュータネットワークという環境新たに、音声を導入してみてはどうだろうか。このような新しい視点から音声利用の可能性を考えよう、というのが本論文の目的である。

本論文では「ネットワーク環境では」という視点でこれら音声の利用の問題を考える。これを実現させるには解決しなければならない問題が数多

く残されている。例えば、音声合成の方法の問題、またネットワーク環境ならではの通信トラフィックの問題などを挙げることができる。この研究を通して、これらの問題点を明確にし更にその解決方法を考えていくことにする。

本論文では、第二章で、音声合成の技法について考察し、第三章でネットワーク環境における音声の利用可能性を考える。そして第四章では phone メッセージの音声化について、そして第五章では、今後の課題について述べる。最後に第六章でこれらをまとめる。

第 2 章

音声合成の技法の発展と現状

2.1　音声を導入する意義

ここではまず、ネットワーク環境での音声利用ということを考える前に、音声そのものの持つ性質、効果について検討する。コンピュータ利用が広がりつつある今日、それを使う人間はほとんど視覚のみに頼っている。この環境に今回、聴覚を用いる音声を導入しようとしているわけである。

情報を伝達する手段として、聴覚を使うことの利点とはなんであるかを考えるため、今まで用いられてきた視覚とこれから用いようとしている聴覚を色々な面から比較する。

- まず最も大切なことである、情報伝達の ”速さ ”と ”正確 ”さで比較してみる。しかし、これは明らかに視覚のほうが上である。「 見る 」という行為は一瞬行なうだけで、物の形、色、状況、などの情報を得ることが出来る。しかも、自分が直接そのものと関わるのだから正確である。これを、聴覚を使って同じだけの情報を得ようと思ったら、大変な労力と時間がかかることだろう。また、「 聞く 」ときは、一度聞いたらもうその情報はもどってこない。一過性のものである。しかし「 見る 」ほうは何度でも確かめられる。このように、視覚を使った方が聴覚を使うより、より正確にしかも速いといえるだろう。昔の諺に「百聞は一見にしかず」というのがある。これは、このことをよく表していると言える。

- 次に情報をより多くの人に伝えたい時は、どうか。この場合は音のほうが便利だろう。しかしこれは逆に、伝えたくない人にまで、情報をもらしてしまうということが起こりかねない。従って、どちらがいいとも悪いともいえない。

- 次に、環境の影響の面ではどうだろうか。「聞く」ということは暗くても出来る。しかし、「見る」ことは暗くては出来ない。また、「見る」

ことは騒々しくても出来るが、「聞く」ことは出来ない。つまり、このときもどちらがいいとも悪いともいえない。

これらの比較からわかる事は、視覚も聴覚も、それぞれお互いにはない有用なものを持っているということである。ただコンピュータの使用ということになると、視覚はそれのみでも十分実用出来る（している）が、聴覚のみではちょつと無理がある。従って今のコンピュータに音声を導入するということは、現在のコンピュータの使用方法を、根本から覆すというものではなく更に使い易いものにするというほどのものである。(将来どのようになるかは、わからないが)

それでは音声には、どのような使い道があるのだろうか。音が持っている性質で、他より優れた点とはなにか。それは、なんといっても人の注意を引くという性質である。これは他とは比べものにならない。例えばデパートでよく聞く「アテンション、プリーズ」という声。あれが聞こえると、ほとんどの人は耳を傾ける。別に、その人が何をしていようと関係ない。その声はあらゆる人の耳にはいってくる。

音声は今、人間同士が用いているの最も大切な情報交換（情報伝達ではなく）の手段である。情報をお互いに伝えあうということでは音声に勝るものはない。また、音声の中には単なる音には無い情報さえはいっている。１つは、発声している人は誰かという情報。更に人間の感情さえ、そこには入ってくる。とうてい、文字を見ているだけでは、このような情報は伝わってこない。このようなことを考えれば、音声をコンピュータ（ネットワーク）の中に取り入れて、情報伝達の手段とすることは、十分意義のあることではないだろうか。

2.2 音声信号のデジタル化

音声信号すなわち音声波形はマイクロンによって電気信号に変えられ、我々が処理出来るものとなる。この電気信号に変えられた音声信号は今日ではアナログ信号のままでなくデジタル信号に変換されてから処理されることが多い。これは非常に多くの理由によるが、主な理由は２つである。まず、１つめはデジタル技術を使う事によって極めて技巧をこらしたアナログ的には実現出来ないような信号処理機能が実現出来ることである。第二にデジタル信号は信頼性があり、しかも非常にコンパクトであることである。数年前から急速に普及したコンパクトディスク（CD）もデジタル信号を使っている。CD は、アナログを用いる従来のオーディオカセットに比べて非常に音質が優れ、また何回再生しても同じ音が期待できる。また最近、半導体技術は大きく進歩し、それに伴って電子計算機の集積回路も急

速な発展を遂げた。更に、デジタル通信であるコンピュータネットッワークも急速に発達しつつある。これらのことは、音声をネットワークに導入するうえで、非常に好都合であると言える。

さて、アナログ信号である音声を、デジタル信号に変換してやるにはどうすればいいだろうか。アナログ信号をデジタル信号に変換することをAD 変換、逆にデジタル信号をアナログ信号に変換してやる事をDA 変換という。AD 変換を行なうには、標本化（sampling）、量子化（quantizing）、および符号化（coding）が必要である。簡単にいえば、標本化というのは物理的に連続な波形を、時間的に離散的な時点の値の系列で表現することである。量子化というのは、波形の値を有限個の値の中の１つで近似的に表現することである。また、符号化というのは具体的にどのように表現するかということで、通常は２進符号化が行なわれる。それでは、それぞれを詳しくみていくことにする。

2.2.1　標本化

あるアナログ信号波形 $x(t)$ があったとする。これは時間的に離散的な時点 $t_i = iT$（$i : integer$）での値の系列、$x_i = x(iT)$ に変換される。ここで T[s] ををを標本化周期（sampling period）と呼び、この逆数 $S = 1/T$[Hz] を標本化周波数（sampling frequency）という。sampring period を大きくしすぎると元の波形が再現できない。また小さくし過ぎると無駄なメモリーを大量に消費する。そこでこれを決定するために、標本化定理が用いられる。例えば、アナログ信号波形 $x(t)$ が、$0 \sim \omega$[Hz] の間に帯域制限されている時、$x(t)$ を $T = 1/2\omega$ ごとに標本化すると次のような式で元の波形が再現出来る。

$$x(t) = \sum_{i=-\infty}^{\infty} x\left(\frac{i}{2\omega}\right) \frac{\sin\{2\pi\omega(t - \frac{i}{2\omega})\}}{2\pi\omega(t - \frac{i}{2\omega})}$$

ここで、$x(\frac{i}{2w})$ は、$x(t)$ の $t_i = i/2\omega$（i：定数）におけるサンプル値である。

だから、例えば人間の耳は 20kHz 位までの周波数を聞き取れるとされているからこの場合 sampring frequency はその 2 倍の 40kHz 以上にすると良い（CD プレーヤーの sampling frequency もこれを参考に決定されている）。また、電話の場合帯域が 3.5kHz 程度であるので、サンプリング周波数は 8kHz 程度で十分である。（今回の実験では、37.8kHz で行なった。）

2.2.2 量子化

波形の振幅を量子化するには、振幅の全範囲を有限個に分割して、その中の一つの範囲に入る波形をすべて同じ振幅値とみなす。例として 8 レベル量子化器の入出力特性を例を示す（Figure2.1）。

ここで、Δは量子化ステップを表す。量子化の特性に関連するパラメータはレベル数と量子化ステップ（Δとする）であり、レベル数は、Bbit の符号化を前提とするとき、2^B 個にするのが普通である（２進符号語が最も効率良く使えるから）。信号の値の範囲が $|x_i| < x_{max}$ と仮定すると

$2 \times x_{max} = \Delta 2^B$

としなければならない。

量子化後の標本値 y_i と元のアナログ値 x_i の誤差 $e_i = y_i - x_i$ を、量子化雑音（quantization noice）（量子化誤差と呼ばれることもある）という。図／／／から Δ と B が上式のように選ばれていれば、量子化雑音は、

$$\frac{-\Delta}{2} \leq e_i \leq \frac{\Delta}{2}$$

であることがわかる。

量子化雑音について次のような統計的モデルを仮定する。

1. 量子化雑音は定常的な白色雑音仮定である。

2. 量子化雑音は入力信号と無相関である。

3. 量子化雑音の分布は各量子化区間に渡って一様であり区間はすべて一様なので、次の式が成り立つ。

$$P_e(e_i) = \begin{cases} 1/\Delta & (-\Delta/2 \leq e_i \leq \Delta/2) \\ 0 & (\text{その他の } e_i) \end{cases}$$

また一般に、信号対量子化雑音比（S/N 比）は B ビットで符号化した場合、ほぼ $6(B-1)$ dB という式でで与えられる。これより、実用上の観点から考えれば 12 ビット符合化のとき S/N 比は 66dB ということになり、これ位あれば十分であろうということがわかる。また、8 ビット符合化なら S/N 比は 42dB ということになり、これ位でもまずまずの音が再現できるということが考えられる。

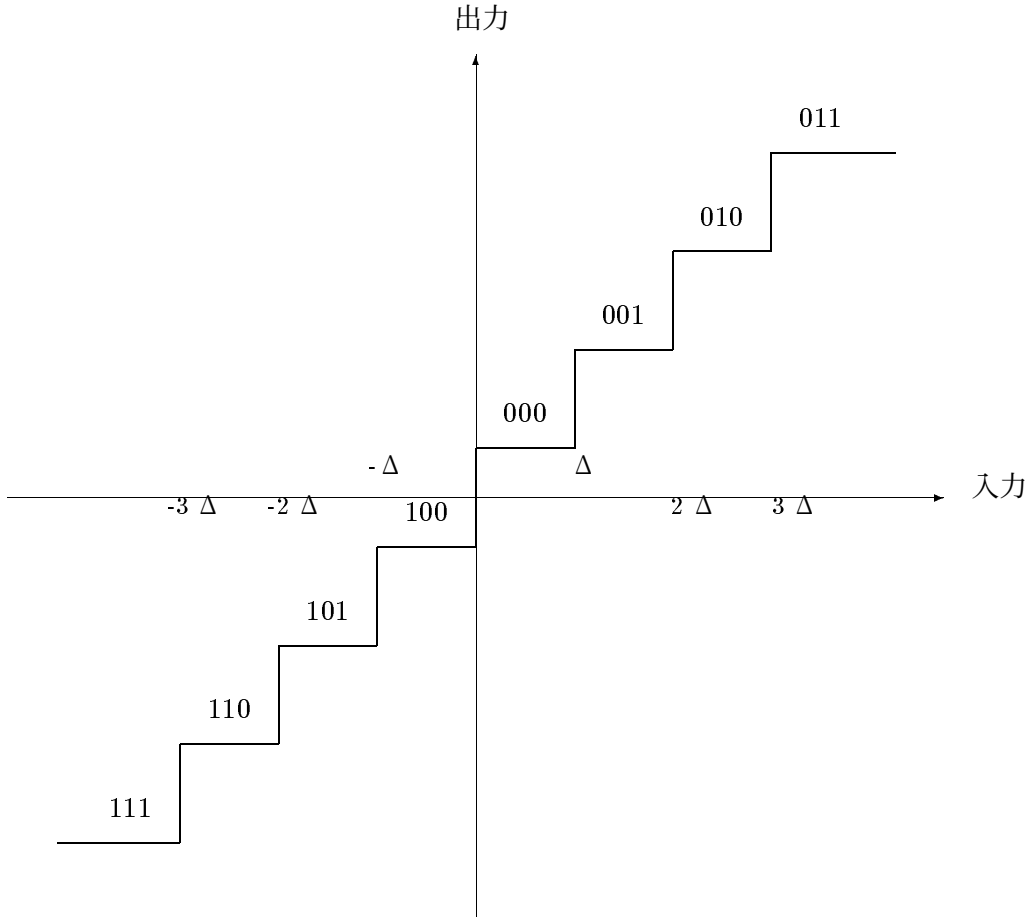


図 2.1: ８レベル量子化の入出力特性の例

2.3 音声信号の符号化

2.3.1 音声信号の符号化とは

音声を電子計算機で扱う場合、音声波形を前でみたようにある周期ごとにサンプリングし、各サンプル点での音声の値を AD 変換して、その値を０と１の符号列で表す。これを符号化というわけである。その中で最も簡単なものが Pulse Code Modulation（PCM）と呼ばれるものである。これはアナログ信号を一様なステップで量子化 するもので、通常 A/D 変換として行われているものと同じである。この方式は情報圧縮をしていない。したがって例えばサンプリング周波数 f と a ビットの量子化では、$a \times f$ が１秒間の音声の蓄積に必要なビット数になる。また、この $a \times f$ は音声を伝達するのに必要とする帯域幅に関係する。例えば、10 kHz で 8bit 量子化すると 10×8 kbps の伝送路が必要となる。信号値をデジタル変換するとき必要なビット数は、どの程度の音質を望むかによって異なってくる。一般に量子化ステップを Δ、原信号の振幅の存在範囲を L とすると量子化ビット数 B は、$\Delta 2^B \geq$ L でなければならない。また、B ビットで量子化されたときの PCM 信号の S/N 比は前にみたように、$6(B-1)$ db で与えられる。したがって、これが原信号の S/N 比 より劣化しないように B を定めなければならない。例えばこの方法だと、4 kHz の帯域を持つ音の量子化雑音を無視出来るほどにするには約 100（kbit/s）前後（8 kHz サンプリング、13 bit 量子化）の情報量が必要となる。

これをみてわかるように、音声をコンピュータで扱う時に問題なのは大量のメモリを消費することである。したがってなるべく音声データを効率的に（少ないメモリで、高質な音がでるように）、蓄積することが大切であり様々な研究がされている。これは音声の重要な研究分野となっており、高能率符号化と呼ばれる。このなかで、今回の研究で関連のあるものを考察してみる。

2.3.2 LOG-PCM

通常電話系に用いられている。これは音声振幅の統計的性質を用いて、振幅を対数変換して圧縮し、その結果を線形に符号化する。音声信号の振幅分布は指数分布を示すので、対数変換することにより各ビットの出現頻度が等しくなり、情報理論から明らかなように、歪みを最小化することが出来る。LOG-PCM の代表的な変換公式には、μ-Law と A-Law の二つがある。これら二つの、間には互換性はない。μ-Law はアメリカや日本で、A-Law はヨーロッパで使われている。μ-Law の変換式は次の式で表される。

$$F(x) = sgn(x) \times \frac{\ln(1+\mu \times |x|)}{\ln(1+\mu)}$$
$F(x)$: output/127
x : inout/32768
μ : constant255

ここで、μ が 圧縮の程度を表すパラメータである。通常 100 から 500 の値が用いられる。

また、 A-Law の変換式は次の式で表される。

(1) $0 \leq |x| < 1/A$ のとき

$$F(x) = sgn(x) \times \frac{A \times |x|}{(1+\ln(A))}$$

(2) $1/A \leq |x| \leq 1$ のとき

$$F(x) = sgn(x) \times \frac{1+\ln(A \times |x|)}{1+\ln(A)}$$

$F(x)$: output/127
x : input/32768
A : constant87.6

2.3.3 ADPCM

適応量子化

音声の動的特性の非定常性に更に有効に対処するため、PCM の量子化器のステップ幅を振幅の rms 値（[90] 参照）に応じて時間的に変化させる方法が用いられており、適応 PCM（adaptive PCM）方式と呼ばれる。このような量子化方法が適応量子化である。音声信号は、短い区間でみれば定常とみなせるので、rms レベル、したがってステップ幅は sampling 周期に比べれば、かなりゆっくり変化させればよい。ステップ幅を変化させる方法には 2 種類あるが、そのうち ADPCM（後述）に関係のあるのは、後向適応方式（逐次適応方式）である。これは量子化して伝達した振幅情報からステップ幅を逐次自動生成するので、ステップ幅を伝送する必要がない。

予測符号化

音声信号は隣接標本化のみならず、更に離れた点の間でも相関がある。このため、隣接標本間の差信号（予測残差）を符号化することにより、情報圧縮を図ることができる。この方式は DPCM と呼ばれる。即ち差分、あ

表 2.1: 音声１秒間あたりのデータ量の比較 （単位は [Byte]）

format \ sampling	8kHz	9.45kHz	18.9kHz	37.8kHz
ストレート PCM(8[bit])	8000	9450	18900	37800
ストレート PCM(16[bit])	16000	18900	37800	75600
ADPCM (4[bit])	4000	4725	9450	18900
ADPCM (8[bit])	8000	9450	18900	37800
μ-Law ／ A-Law(8[bit])	8000	9450	8900	37800

るいは予測残差は原信号の標本値の分布に比べて、変化範囲の平均エネルギーが小さいので量子化ビットが少なくて済む。この原理に基ずく方式を予測符号化と呼ぶ。

ADPCM

これは前の適応量子化、予測符号化の両方あるいはいずれかを用いる DPCM のことである。この方式によると、32 kbps（8 bit サンプリング、4 bit 量子化）で、約 22 dB の S/N 比が得られ、log-PCM より約 8 dB の改善が得られる。また、4 bit の ADPCM は 6 bit の log-PCM と 7bit の log-PCM の中間にあると評価され、平均約 2.5 bit の改善がある。ADPCM に変換すると、16bit ストレート PCM データを最大 2/7 まで圧縮出来る。

2.3.4 圧縮方式の比較

以上述べてきたような３つの音声符号化方式が今回の研究に関係するものである。これら３つの方式を比較した結果を以下に示す（Table2.1、Table2.2、Table2.3、Table2.4、Table2.5参照）。

2.4 現在の音声合成

2.4.1 音声合成の原理

音声を直接人間の発声によらないで、人工的に作り出すことを音声合成という。一昔前までは、これらを実現させるための合成器は機械的構造のものばかりだった。記録として残っている最初のものは１７９７年に作られたといわれる。近年の電子計算機、コンピュータネットワークの著しい進歩に伴い音声合成の技術は具体的な適応領域をもつようになった。ま

表 2.2: 1 [MB] あたりの音声の時間の比較 （単位は [秒]）

format \ sampling	8kHz	9.45kHz	18.9kHz	37.8kHz
ストレート PCM(8[bit])	131	111	55.5	27.7
ストレート PCM(16[bit])	65.5	55.5	27.7	13.8
ADPCM (4[bit])		222	111	55.5
ADPCM (8[bit])		111	55.5	27.7
μ-Law ／ A-Law(8[bit])	131	111	55.5	27.7

表 2.3: ダイナミックレンジの比較

format	range
ストレート PCM(8[bit])	可
ストレート PCM(16[bit])	良
ADPCM (4[bit])	良
ADPCM (8[bit])	良
μ-Law ／ A-Law(8[bit])	良

表 2.4: ノイズの多少の比較

format	noise
ストレート PCM(8[bit])	中
ストレート PCM(16[bit])	少
ADPCM (4[bit])	多
ADPCM (8[bit])	中
μ-Law ／ A-Law(8[bit])	中

表 2.5: サンプリング周波数と音質の関係

sampling	quality
8kHz	電話並み
9.45kHz	電話並み
18.9kHz	中音質
37.8kHz	高音質

た線形予測分析法の導入が「てこ」となって研究が著しく進歩した。音声合成の方式は次の３つの方式に分類されている。

- 録音編集方式 · · · 人が発声した音声波形をそのままあるいは波形符号かして蓄積しておき、必要に応じてつなぎ合わせて使う方式。
- パラメータ編集方式 · · · 人間の音声を直接利用せず、音声のパラメータだけを抽出し人間の声道モデルと組み合わせ、合成音を発声する方式。
- 規則編集方式 · · · 文字列あるいは音素記号列から、音声学的、言語学的機能に基づいて音声を作り出す方式。

録音編集方式によるものは、今日広く実用されており、パラメータ編集方式も実用化段階に入っている。

2.4.2 録音編集方式

あらかじめ人が発声した音声を、単語や文節などを単位に取って蓄積しておき、必要に応じてそれらを読み出して接続し、音声を合成する方式である。この方法による音声合成は、アナログ録音では非常に困難である。なぜならこの方式では、必要とする音声信号をすべて録音し蓄積しなければならず、大量のメモリや、必要とする音声信号を高速に呼び出す制御回路が必要なためである。現在の半導体技術の向上や、入出力装置の高速化、大容量化の進歩によって初めて実現可能となった。デジタル記憶装置に記録した音声は何回再生しても音質が低下することが無く、また単語や単文というこまぎれの音声のつぎはぎが、自由に出来る。現在では、比較的少ない単語などの組合せ編集ですむ音声の情報入出力用として、すでに多くの所で実用化されている。この方式は実用的であり、電話を用いたサービスや駅や空港の自動アナウンスシステムに利用されている。

この方式は、記録しておいた音声をつぎはぎするものである。したがって、つなぎ合わせても不自然なアクセントにならないようにすることが、最大の技術的課題である。そのため録音の単位をどうするかが問題になる。蓄積する単位を文節、文というように大きくとればとるほど、作り出される音声の品質は良くなるが、合成出来る語彙や文章の種類は限られる。一方蓄積単位を単音のような小さいものにすると、合成できる語彙や文章の種類の自由度は、増すが、音声の品質が著しく低下する。

そこで、必要とするメモリ容量を小さくするために、単語よりも小さい単音節や VCV 連鎖（C ： 子音、V ： 母音）などを使う方法が考えられる。例えば日本語の場合、５０音に相当し、消音と呼ばれる５０音に、濁音、促音（ッ）、拗音などを合わせて合計１００種類程度の単音節を組み合わせれば理論的には任意の文章を合成出来るわけである。しかし、この音節による編集合成音声は、その音質が良くない。

この理由は、連続音声中の音声信号は前の音声の影響をうけ、その状態が変化するためである。そのため同じ音節でも単独に発声されたものと、連続した単語中に出てくる音節は、かなり違ってきてしまう。しかし、この音節単位の方式は日本語の場合 0.1 秒から 0.2 秒で発声され、計算機で取り扱う単位として音声認識を行うにも適当な大きさである。この方式は、単語数に制限を受けることなく、アルファベットまたは仮名によるテキストをいれると、連続音声を出力させることが出来る。

音質を良くするための簡単な方法は、録音の単位を出来るだけ長くすることである。例えば自動アナウンスシステムでその内容全てをあらかじめ録音しておけば、アナウンサーの声と少しも変わらない音質で放送出来る。しかし、それでは大きなメモリを消費することになり、コストも増大し、必要なメモリへのアクセス時間も無視出来ない。

したがって、実用的には単語とそのつなぎに使う短い文章を記憶させておく。その時も本当に人が話すのに近い自然な音声にしようとすれば、単語やつなぎの文章には、場合場合によって異なるアクセントが要求される。したがってそれが必要な時には、尻上がり、尻下がり、平坦などのアクセントの音声を録音しておき、適当なものを選んで使わなければならない。

この方式の問題点をまとめてみると次のようなことである。まず、語彙が限られること。そのため実用的なメモリ容量ではアナウンスの内容が限定され、任意の文章を発声させるようなことは出来ない。したがってあらかじめ計画された範囲内でしか使用できない。コンピュータのエラーメッセージ程度なら語彙の制約の問題は現在の高速でしかも大容量のメモリで解決出来る。しかし、この方式は致命的な欠点がある。すなわち、１つのメッセージとしてまとめるには、すべての単語が同じ発声者の声でなけれ

ばならない。別の人の声を混ぜると奇妙なものとなってしまう。よって、すべての単語を１人のアナウンサーに発声させてメモリに蓄積しておかなければならない。最初から十分な計画を立てて蓄積すべき単語を選ばなくてはならない。それでも完全ではない。新しい単語を加える必要が生じたとき、同じアナウンサーを捜し出してしかも昔と同じ声を出すようにしなければならない。これは、困難なことである。（それが１０年も２０年もたった後ならなおさらである）このような場合に対処するため、音声の音質を他人の声と同じに聞こえるように変換する技術の開発も音声研究のひとつの目標になっている。しかし現在ではまだ、不可能といってよい。

2.4.3 パラメータ編集方式

この方式は録音編集方式と同様に単語や文節などを単位とする。しかし、録音編集方式のように人間の声を直接利用せず、音声のパラメータだけを抽出し、人間の声道モデルと組み合わせて合成音を発声する方式である。波形を蓄える場合に比べて、合成音声の自然性は方式により若干低下するが大幅な情報圧縮がはかれる。更にパラメータを抽出することにより、時間長の伸縮や接続部のピッチやスペクトル変化の平滑化などを行う事ができる。具体的な音声合成器として、チャネルボコーダや LSP、PARCOR 方式などの線形予測分析法に基づく合成器が用いられる。

2.4.4 規則合成方式

蓄えておく単位として音節、音素、１ピッチ区間の波形などのような、基本的な小さな単位の特徴パラメータを用い、その変わりそれらを接続する規則や、ピッチ、振幅などの韻律情報を制御する規則を精密に定めることにより、いかなる言葉でも、音素、音韻記号あるいは文字の系列から合成出来るようにしようとする方式である。合成に用いられる音声の最小の単位を基本単位と呼ぶ。合成単位としては音素が基本である。これだと３０から５０種類ですむので記憶容量は少なくてすむが、この結合規則はかなり複雑で良い品質を得るのが難しい。このため、これよりやや大きい単位を用いることが多い。日本語の場合は仮名文字に対応する１００音節が用いられることが多い。より良質な合成音を得るために CVC（C：子音、V：母音）を単位とする方式も検討されている。日本語の場合可能なすべての CVC を準備すると５０００から６０００種類にものぼる。そのため出てくる頻度の高い約１０００種類の CVC と VC 単位が用いられている。VCV 単位を用いる方法も検討されている。この場合は７００から８００種類ですむ。例えば「 とよ 」という単語は、CV 単位では to + yo、CVC 単位では toy + yo、VCV 単位では to + oyo となる。英語では音節の種

表 2.6: 音声合成の比較

項目	録音編集方式	パラメータ編集方式	規則合成方式
音声の了解性	高	高	中
音声の自然性	高	中	低
語い数	少（500 語以下）	大（数千語）	無限
情報量	24 〜 64kbps	2.4 〜 9.6kHz	50 〜 75bps
１ Mbit での音声長	15 〜 40s	100s 〜 7min	無限
音声蓄積単位	単語、文節、文章	単語、文章、文節	音素、音節など
装置	簡単	やや複雑	複雑
ハードウェア主体	記憶装置	処理と記憶の併用	処理装

類が３５００種類以上、異音を考慮すると約１００００種にものぼる。このため、音節を分解し、２音節の組合せ（CV、CVC など）を基本とした、dyad、diphone、（両者共約４００ から １０００種）、これらよりやや大きい単位の demisyllable（約１０００種）などの単位が用いられることが多い。

日本語の場合について言えば、単音節程度の基本単位でも十分である。ただし、音節単位を連結する場合、自然かつ明瞭となるように音声情報を編集し直して連結する必要がある。また単語や会話音ではイントネーションやアクセントが重要な役割をもっている。イントネーションはピッチ周波数と関係しており、合成の際にそのピッチに関する規則が必要である。

最後にこれら３つの音声合成の比較をしてみる。（Table2.6参照）

第 3 章

ネットワーク環境への音声の導入

２１世紀は今以上にコンピュータを利用する機会が増え、また世の中に広まっていくことは容易に想像がつくことである。ところで、今までコンピュータはどのように発展してきたのであろうか。大まかな図を書くとすれば次のページのような図が書ける。

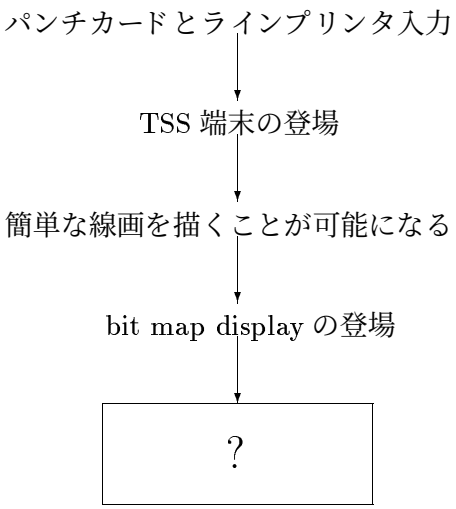
そしてこの図の一番下の四角にいれるべき文字が、これから行なわれるであろう「マルチメディアの導入」の１０文字である。今後、さまざまな形で、マルチメディアの導入がされることになる。この一つの形態が今回の研究の課題である音声の導入なのである。

一方、コンピュータをつなげる研究（コンピュータ ネットワークの研究）も日々着実に進歩し続けている。そして２１世紀には、これら一連の流れに今までとは少し違った新しい流れが加わる。そこに加わる新たな流れとは、これらの統合化のことである。２１世紀は、マルチメディアネットワーク時代の幕開けなのである。もちろん、個々の問題だけをみつめて解決してくことも大切である。しかしこれからは、統合化というもっと広い視野で問題を見つめ解決していくことが大切になってくる。

その第一段階として、今回音声をネットワークに導入することにしたのである。そうすることによって、何が起きるのか、何が問題なのか、それをどう解決していけばいいか、ということがはっきりと認識することが出来るようになる。そういう意味でこの研究は音声そのものの研究というわけではなく、ネットワーク環境におけるそれの持つ有意性を考えるための研究、さらにもっと視野を広げれば、来るべきマルチメディアネットワーク通信時代の新しい流れを考察するための研究なのである。

3.1　ネットワークへの音声の導入の意義

前に音そのものの持つ性質、また、音をコンピュータに導入する意義については考えてきた。今度は、これをネットワークという環境に取りいれた場合について考えてみる。



例えば今、２つのモデルを考える。それらを、個室モデルと教室モデルと呼ぶことにする。個室モデルというのは、個室のなかで自分１人だけが、ワークステーションで仕事をしているような状態である（Figure3.1）。

教室モデルというのは、教室で勉強している時のように大勢いる中で仕事をしているような状態のことを表すとする（つまり、すべての人が音が聞こえる状態である）。また、教室モデルのようでも、音が聞こえる状態にあるのは自分だけという時は、大勢の中で仕事をするような形態でも個室モデルと考えることにする（Figure3.2参照）。

これら２つ場合を比較してみると、音のもつ人の注意を引くという性質はどちらにも有効に働く。しかし、何を音声として出力させるかによってその効果がことなってくる。例えば個室モデルで、エラーメッセージを音声で出力しているとしよう。もし、これを教室モデルで行なっていたら何が起きるであろうか。おそらく、それを使用している人以外は「うるさい音だ」と思うであろう。また逆に、もし教室レベルで phone がかかってきたとき、それを音声を使って教えてくれるという機能があったとする。これを、個室モデルで使ったらどうなるだろうか。自分が呼ばれている時ならそれは有効であるが、他人を呼ぶものだったらあまりそれは意味がない。このように環境によって、その音声が有効に働くか逆に騒音になってしまうかが異なってくる。

まとめると、ネットワーク環境において音声を使用する場合音声は次の２種類に大きく分けられる。

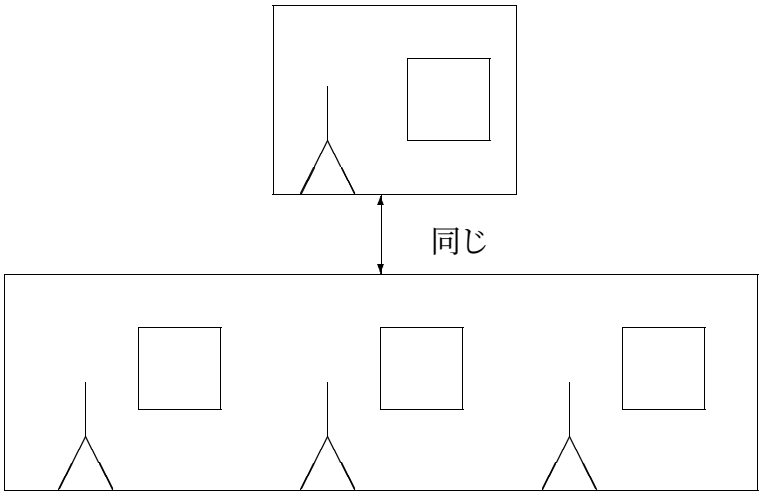


図 3.1: 個室モデル

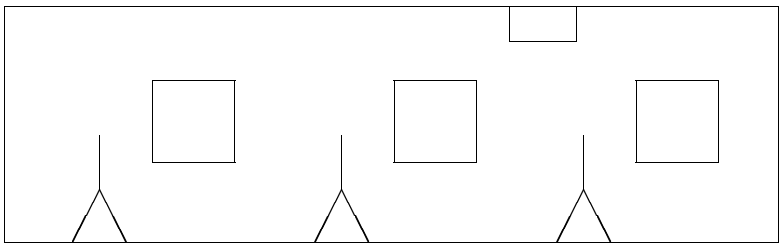


図 3.2: 教室モデル

- 個室モデルでしか使用しないもの。
- 主に教室モデルで使用するもの。

これら２種類を、はっきりと区別することは難しいが、これらのデータをうまく使い分けないと、音声はただの騒音になってしまうこともありうる。

本研究で選んだのは上の例で述べた phone メッセージの音声化である。これを、選んだ理由の１つとして、個室レベルでしか利用可能性のないものは、それを使用する１人にしかメリットがない。したがって、まだ音声を使える機械が２、３台しかない今の段階では時期尚早であることということがあげられる。（その他にもあるが、それは後述することにする。）では、これを実現するにはどのような問題が潜んでいるだろうか。大きく分けて二つの問題がある。一つはデータベースシステムの設計の問題であり、もう一つはデータ通信上の問題である。これは音声が多量のデータを必要とすることに起因する。そこで、これらの問題について次に続く二つの節で考えていく。

3.2　データベースシステムの設計

初期のデータベースシステムは適用業務プログラムのバッチ処理により、データベースが共有される形態でシステムが運用されていた。そして、コンピュータ利用の普及により、データベースのオンライン利用が行なわれ、データベースは遠隔の端末からも利用されるようになった。現在では、更にその利用が拡大している。それに加え最近では、データベースのオンライン化、ネットワークの発達に伴い、データベースをネットワーク上で利用できるシステムが必要となってきている。**WIDE**（Widely Integrated Distributed Environment）環境に音声を取り入れるためにも当然に、このシステム（分散データベースシステム）の開発が必要である。本節では、どのようにこのシステムを取り入れていったらいいだろうか、ということについて考察する。

3.2.1　分散型データベースシステムとは

分散型データベースシステムとは、物理的に分散した複数のデータベース管理システム内のデータベースを、複数の利用者があたかも集中型データベースシステム内のデータベースシステムのように利用出来るような機能を提供するシステムをいう。ここでは、分散型の特質を、集中型と比較しながら考えてみる。

1. 信頼性の問題

   分散型データベースシステムは、分散した複数のデータベースシステム管理システムを持っている。したがって、その中の一つがもし故障しても残りのものを用いれば、それをカバーできることになる。勿論故障すればその分サービスレベルは低下するが、それが全く出来なくなることはないので、信頼性は向上する。

2. コストの問題

   データベースを利用する側としては、データが全部身近にあるほうが情報を得るのにかかる時間も少なくて済むし、費用（通信のための）もかからない。しかし、それでは各地域ごとにすべてのデータを持っていなければならず、実際的ではない。したがって、地域ごとに頻繁に用いるデータだけをそれぞれに持たせておき、あまり普段は必要としないものだけを他のデータベースから取ってくる。という形態の方が効率が良い。まとめると、データベースの利用に地域的なかたよりがある場合、分散データベースシステムは、うまく設計すればコストを大幅に削減出来るかもしれない。

3. 拡充性

   分散型データベースシステムは、集中型データベースシステムに比べて拡充性がある。もし、現在のデータベースシステムでは容量や能力が不十分であれば、データベースを増築すればいい。分散型だと、集中型に比べてそれが容易に出来る。

4. 格納場所の独立

   データベースの使用者は、そのデータベースが通信網上のどこにあるかを知らずにそれを利用出来る。

このように、分散型データシステムは集中型データベースシステムでは実現困難なものも、提供できる可能性がある。従って、これからのネットワーク時代にはこの技術はぜひとも必要不可欠なのである。

3.2.2 システム形態

分散型データシステムのシステム形態を考えるためには、その物理的な構成要素と論理的な分散形態の二つの側面を見る必要がある。まず、初めに物理的な構成要素がどうなっているかをみてみる。

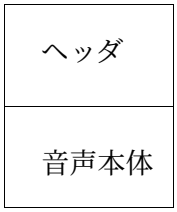


図 3.3: 音声ファイル全体のフォーマット

物理的構成要素

分散型データシステムで使用されるコンピュータとしては、データベースの容量、利用形態、コストなどの面から、大型ホストコンピュータ、ミニコンピュータ、ワークステーションなどがあげられる。このうち、今回の研究に用いたものはワークステーションである。これは、ローカルな共有データベースをもち、小規模な分散型データベースシステムの中核的構成要素となり、勿論音声を使用することが可能である。(データベースのこととは直接関係はないが、今回用いた機材では音声を一つのファイルとして扱える。従って、一般のデータファイル（音声以外のファイル）のデータベースを考える方法と同じように音声データのデータベースの構造を考えることができる。なお、音声ファイルのフォーマットおよびヘッダフォーマットを図に示す（Figure3.3、Table??参照)）。

また、大型ホストコンピュータは、大量のデータベースを格納し、大規模な分散データベースシステムの構成要素として中核的機能を果たすことが出来る。もし、音声の利用が進み大量のデータが必要になった折にはこれを利用すべきであろう。もう一つ物理的な構成要素どして通信網の形態が考えられるが、これは後にまわすことにする。

論理的構成要素

次に、論理的形態について考える。これは、どこに、何を音声として採り入れるかによるが、ここでは実験の目標である、**WIDE** 環境に、phone メッセージを音声として取り入れるにはという視点から考える。この点から考えてみると最も適しているのは、すべてのデータベースサイが対等な機能を持つ水平分散形態でろう（Figure3.4参照)。

水平分散形態とはどういうものかということについて、少し説明を加え

表 3.1: ヘッダフォーマット

メンバ名	オフセット	型	内容	長さ（バイト長）
magic	0	char[16]	'@!sound'	16
version	16	int	1	4
offset	20	int	音声のオフセット	4
size	24	int	音声のサイズ	4
block size	28	int	音声のブロック長	4
sb param	32	struct sbparam	音声のフォーマット	28
拡張データ	60	（？）	各種付加情報	不定長、１個以上
パディング	拡張データの直後	char[]	不定	（？）

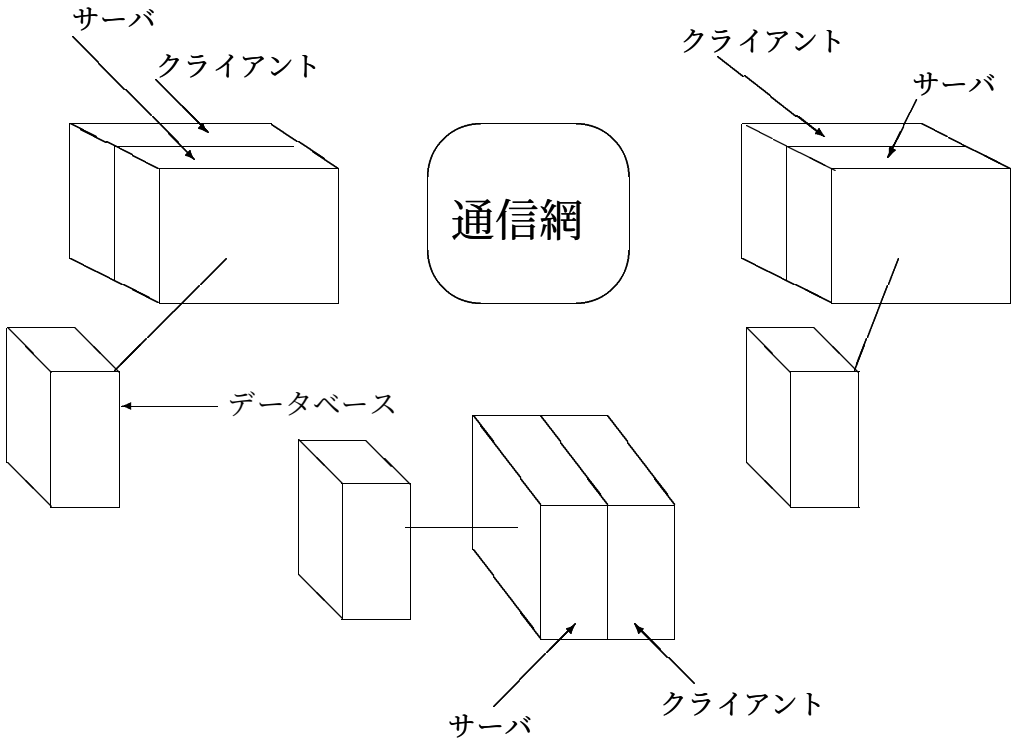


図 3.4: 水平型分散形態

る。水平分散形態において、これを構成するコンピュータの物理的大小は関係ない。コンピュータの提供する論理的機能が対等であるということが問題なのである。この論理的機能は分散型データベースとして要求される基本的機能を想定する。水平分散形態においては、各々のデータベースサイトが、互いに他のサイトのデータベースを認識しており、分散型データベースシステムにおける、クライアントとしてもサーバとしても用いることが出来ることになる。したがって、どのデータベースサイトも他のデータベースサイトのデータベースを利用することが出来るし、他のサイトにデータベースを利用させてやることも可能なわけである。また、各サイトにあるデータベースはシステム全体でデータがどこにあるかとは無関係に共有される。何故これが最も適しているのか。音声データは大量のメモリを必要とする。したがって、その分一般のデータよりも、トラッフィックや、データベース管理、通信費用の問題などが深刻である。また、今後の音声利用の拡大のことなどを考えるとこれが最も適しているのである。

具体的にどのようになものにするかといえば、phone メッセージのなかで、必ず必要な音声データだけをは各大学に保持させておき、大学ごとに特有のデータだけをそれぞれに持たせておくという形態である。もし、自分の大学にない音声データが必要となったときは、そのデータを保持しているデータベースからそれを持ってくる。とはいっても、これはあくまで最終的な目標である。したがって、まづはデータベースを中央サイトで集中的に管理し、遠隔からこの中央データベースを利用するシステム形態を作ることが先決である（Figure3.5参照）。

そして次の段階として、垂直分散形態を少し変形させたもの（一応垂直分散形態をしているが、主側データベースサイトから従属側データベースを利用することもでき、また従属側データベースサイト間で直接データベースを互いに利用しあったりも出来るような形）を使うようにすべきである。このように段階を踏みながら、段々と水平分散形態に近づけていけばいいのではないだろうか。

3.2.3　資源管理

上でみてきたように、まず初めの段階ではデータベースを中央サイトで集中的に管理し、遠隔からこの中央データベースサイトを利用するシステム形態を作ることを目標にする。しかし、注目すべき問題が一つある。今の段階ではまだ問題にならないが、このシステムの規模が大きくなり、分散化が進んでくると、資源管理という問題が浮かび上がってくる。分散型データベースシステムは、通信網で接続された各データベースサイトに、あるデータを共通の資源として管理することが目的である。これを実現す

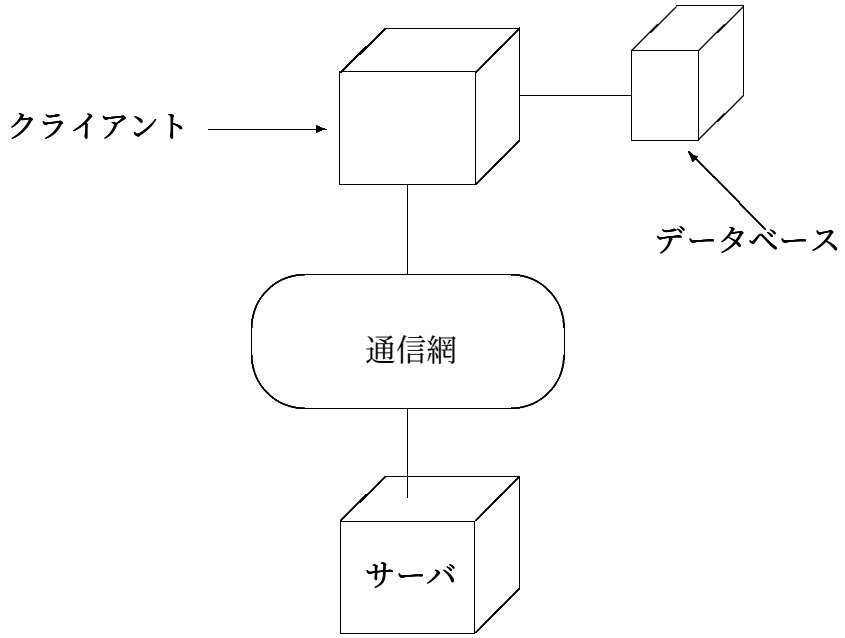


図 3.5: 遠隔データベース利用形態

るには、大きく分けて二つの問題がある。

1. データディクショナリ／ディレクトリ管理の問題

   分散型データベースシステムにおいて、分散して格納されているデータベースに関するデータディレクショナリ／ディレクトリ管理問題をどのデータベースサイトに持たせて管理するかという問題である。例えばもし各データベースサイトすべてのサイト（各大学）にある音声データを完全に重複して持たせているとしよう。この場合データは大学ごとにあるわけであるから、当然自分自身のデータベースから利用したい音声データを手に入れることができる。従って、特定のデータベースサイトに負荷が集中する問題や特定のデータベースサイトが障害となってシステム全体のサービスが停止するという問題は無い。しかし、もしその音声データを変更する必要が生じてきた場合は、すべてのサイトのデータを変更しなくてはならない。こういった問題がある。

2. 命名方式の問題

   分散型データベースシステムでは、各種の問い合わせ処理を実行するためには、データベースを正しく指定できなければならない。ではそれを実現させるには、分散型データベースシステムの名前ずけ方法はどうしたらいいかといった問題である。つまり、集中型命名方式（分散型データベースシステムで取り扱う名前を重複がないように、集中的に管理してやる方式）をとるか、分散型命名方式（データベース管理者が、自分の管理する名前には重複がないようにし他のことは考えない方式）をとった方がいいかということである。

   これら二つの問題は、音声を使って何をさせるかによって、その答が違ってくるであろうから、答を今すぐには出せない。しかし、今回の研究ではシステムは小さいものであるし、現段階では問題定義だけにとどめておく。

3.3　通信網の形態

ここでは、通信網の形態について考える。これも、物理的構成要素の一つではあるが、特に大切なので新しく節を設けた。

一般に、広域通信網には専用回線と公衆網がある。専用線網は、データベースを利用するトラフィックが大きく、利用が激しいような場所で分散が他データベースシステムを構築するのに適している。これには、一般

に 100 bps 程度のものから、数 Mbps の高速のものまであり、最適なものが選択できる。

それでは、**WIDE** 環境においては、どのような通信網の形態になっているかというのを図に示す（Figure3.6参照）。研究が進んでいくにつれて、この環境のすべてに音声が導入されるという日もいつかはくるであろう。しかしながら、今回の研究はまだ実験段階であるし、機材もそれほどまだ整っていないので、一度にこの環境に音声を導入することはしない。とりあえず、身近な環境で実験を行うことにする。その環境として、東大と東工大を選んだ。

今後、この通信網の形態は勿論変わっていくであろう。最近では、新しい通信網として ISDN（サービス総合デジタル網）と呼ばれるものや、衛星通信網などがある。衛星通信網は広域的でありかつ同報通信機能を有しているため、複数のデータベースを広域通信網を介して処理する分散型データベースシステムには適した通信網といえる。また、ISDN は、広域網で LAN と同程度の伝送能力（192kbps 〜 数十 Mbps）をもち、これも有望視されている。**WIDE** 環境においても、この ISDN の研究が日夜進められており、これが導入される日もそう遠いくはないと思われる。

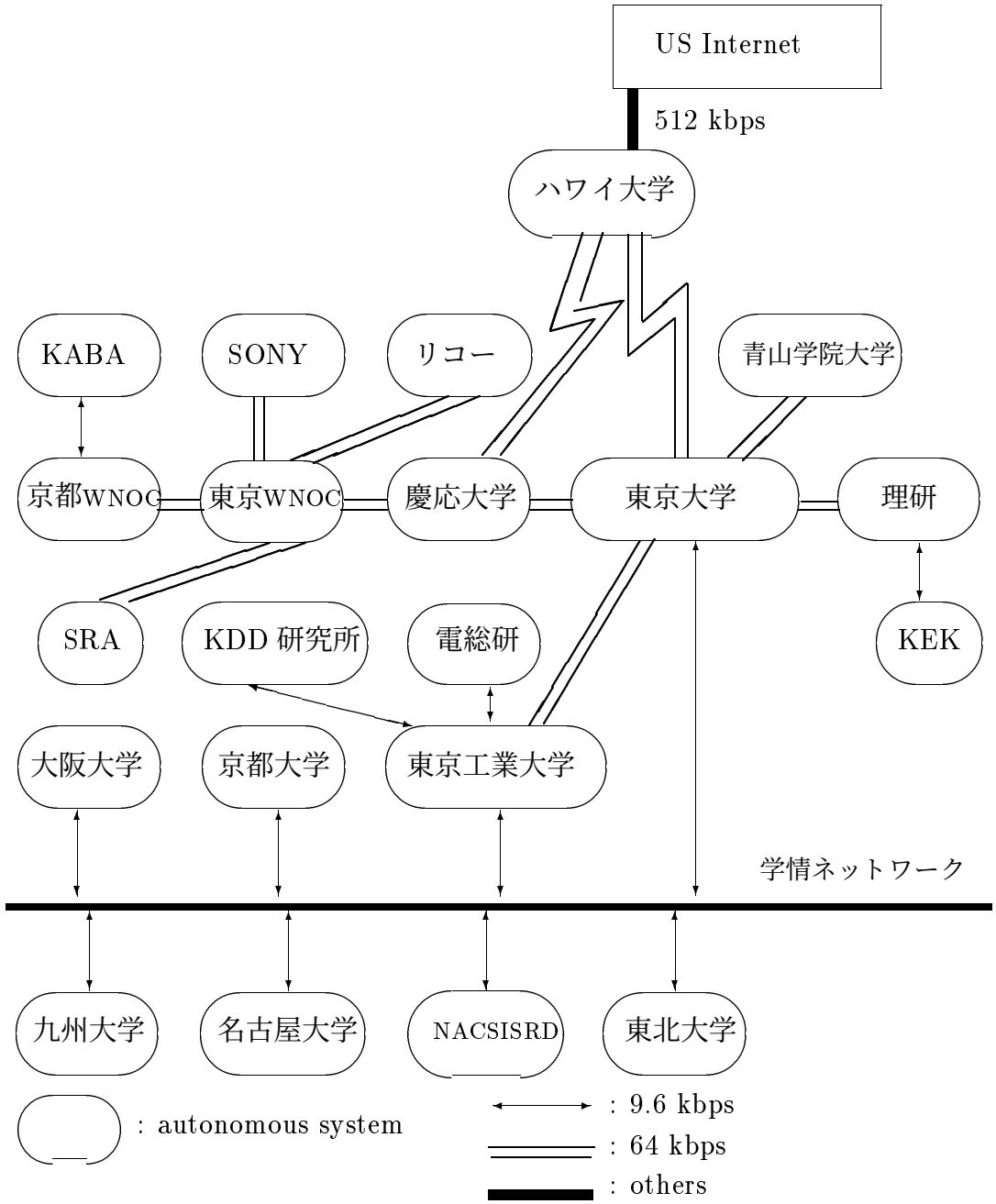

US Internet
512 kbps
ハワイ大学
KABA
SONY
リコー
青山学院大学
京都WNOC
東京WNOC
慶応大学
東京大学
理研
SRA
KDD 研究所
電総研
KEK
大阪大学
京都大学
東京工業大学
学情ネットワーク
九州大学
名古屋大学
NACSISRD
東北大学
: autonomous system
: 9.6 kbps
: 64 kbps
: others

図 3.6: 通信網の形態

第 4 章

音声合成系のネットワーク環境へ導入

4.1 phone メッセージの音声化

音声をネットワーク環境に導入しようという訳だが、一体何を音声化したらよいだろうか。エラーメッセージの音声化、メールの音声化、phone の音声化..... 様々な利用形態が考えられる。さて、そんな中で今回研究課題として選んだのは、音声呼び出し機能付き phone（phone メッセージの音声化）である。これを選んだ理由を以下に述べる。

現在、phoneがかかってくると何が起きているであろうか。これには二つの状態が考えられる。一つは、音も聞こえてくるし、なおかつ画面にはメッセージが表示されているという状態。そしてもう一つは、音は聞こえてくるが、画面には何も見えないという状態。このうち一つめの状態には、別段問題はないであろう。それらだけで、phone がかかってきているということが、十分ユーザに伝わっているからである。しかしもう一方の状態、つまり音しか聞こえない状態の時には、問題がある。この問題は、自由にウィンドウがいくつでも開ける、ということが原因となっている。これは、メッセージを表示しているウィンドウが、自分の今見えているウィンドウの下に隠れてしまっている、という状態に陥っているのである。こんな時ユーザは、音のみでもそれが phone であるということに、遅かれ早かれ気が付くであろう。しかし、それではあまりに不親切なものであるといえる。

これを解決するには、二つの方法が考えられる。一つ目の方法は、始めの例と同じように、必ず自分の見ている画面上にメッセージが表示されるように改良するという方法。そしてもう一つの方は、この環境に音声を導入するという方法である。どちらでも可能な訳であるが、今回の実験では、音声を使ってみようというわけである。もう少し具体的に説明してみる。つまりやろうとしていることは、今の単なる「ピッ」という電子音ではなく、「○○さんから、○○さんへ、phone がかかっているよ」ということを具体的にしゃべらせるような機能をシステムにつけてみようというわけである。これを行うことによって、phone がかかってきているというこ

とを、今以上により確実にユーザに知らせることができるようになるであろう。これが、このテーマを選んだ主たる理由である。

しかし、この研究の持つ意義はそれだけではない。今後２１世紀はいろいろな意味で統合化が進むことが予測される。ネットワーク環境へのマルチメディアの導入というのもその一つである。しかし、それを実現するためには、今後解決しなくてはならない問題が数多く残っている。今回の研究である phone メッセージの音声化は、マルチメディアの導入という作業のほんの一形態でしかない。とはいうものの、それを研究することによって、それを実用化するために何が必要か、また何を解決していけばいいか、ということより明確にすることができる。こういう理由から、今回この研究のテーマを決定した。

4.2　phone メッセージの構造

音声呼び出し機能つき phone をつくるため、音声データベースを作らなくてはならない。この構造を決定するにあたって、一つ頭の中にいれておかなければならないことがある。今回作ろうとしているのは、前にも述べたように、音声呼出だし機能つき phone である。しかしこれは近い将来、このシステムを拡張しあらゆるメッセージを音声化することが出来るようにしたいと考えている。したがって、音声データの構造を考える時には、将来において、今回作るシステムの機能の拡張や変更が、なるべく容易に行えるようなものを今から考えておくべきでなのである。

勿論そのようなことを考えずに、すべての音声データのファイルを一つのディレクトリに作ったとしても、それは設計者の自由であるし、今の段階ではそれほど困ることはないであろう。しかし、システムの規模が大きくなるにつれてそれでは通用しなくなることは目に見えている。(データの整理が困難であるということは勿論、音声ファイルへのアクセス時間がかかりすぎて全く役に立たない。といったことも起きるかもしれない）そこで、このようなことにならないよう、初めから計画的にデータの整理の方法（データベースの構造）を考えることが大切である。音声呼び出し機能つき phone を実現するために必要となる音声データは、個人名と、ほんのわずかな決まりきった単語しかない。しかし、phone メッセージを音声化、更に、あらゆるメッセージを音声化するとなると、これは相当な数にのぼるであろう。

それでは、phone メッセージ構成はどのようになっているかを分析してみよう。phone メッセージをみてみるとわかるが、これらのメッセージは大きく分けて四つのグループに分類することができる。その四つのグルー

プとは次のような四つである。

1. 決まった形をしているもの

   例 · · ·”Warning : no more users can join this conversation! sorry”

2. OS name space（ files, host, users, group など ） と組み合わせて用いるもの

   例 · · ·”Message from phone conversation daemon @ %s”

3. 数字と組み合わせて用いるもの

   例 · · ·”insert entry :%x”

4. その他

   例 · · ·”%c%c%c%s”

更にこれらを詳しく見ていくために、各々のグループ二属するものが、それぞれ phone メッセージ全体中でどれほども割合を占めるかを調べてみた。その結果が以下のようなものである。

- １の形をしたもの.....22.0 %
- ２の形をしたもの.....23.2 %
- ３の形をしたもの.....19.5 %
- ４の形をしたもの.....34.2 %

次に、shell のメッセージはどのようになっているかということも同じように調べてみた。その結果 phone メッセージと同じような四つのグループに分類出来ることがわかった。そこで、更に phone メッセージの場合と同様な方法で、それぞれのグループが shell のメッセージに占める割合を調べてみた。ただし、この場合、全部を調べるのでは、時間的に無理があるので、/usr/src/bin, /usr/src/usr.bin, /usr/src/ucb の三つのディレクトリの下からランダムに百個のファイルを選びそれについてのみ行ってみた。その結果は、以下のとおりである。

- １の形をしたもの.....45.1 %
- ２の形をしたもの.....23.4 %
- ３の形をしたもの.....15.0 %

- ４の形をしたもの.....16.6 %

これらの結果からより、phone メッセージのみの統計結果と、全体としての結果では多少異なるところがあるものの、メッセージは、ほとんど文字（決まった言葉）、数字、OS name の三つの構成要素から成るということがわかってきた。そこで、この結果を考慮にいれた上で、phone メッセージの音声化のための、音声データベースの構造を考えていくことにする。

4.3　音声化の方法

上で見てきたように、メッセージは、文字（決まった言葉）、数字、OS name のほぼ三つで成り立っている。従って、phone メッセージを音声化するための音声データの構造を考える上においても、これらの特徴を生かしたものにするのが望ましい。そこで考えたものが、以下に述べる方法である。（ただし、OS name のうちで、今回は音声呼び出し機能を実現するための、個人名（login name）について、特に考えていくことにする。）

まずメッセージを構成する単語を、定型、数字、OS name の三つのグループに分類してそれぞれグループごとに音声データを蓄積しておく。そして、音声 v 化する際に、それらのグループから該当するデータを検索し、それらをつなぎ合わせて一つのメッセージとして音声化するという方法である（Figure4.1参照）。例として、phone メッセージ中の次のような一文を音声化する方法を考えてみよう。

”Message from phone conversation damon @ %s”

この場合、Message from phone conversation damon @ までは、決まりきった形である。そこで、定型の音声データが入っている場所（ディレクトリ）から、message、from 、phone 、conversation、damon、@、などの音声ファイルをそれぞれ、探し出しそれを音声化する。また %s の中には この場合 host 名が入る。そこで、この部分を音声化するために、OS name 用のディレクトリから、該当する音声データを捜し出してきて、これを読ませるのである。また、ここでは、出てこなかったが、数字の場合も考え方は同じである。

また、今回の実験では前に述べたように、OS name のうち特にユーザの login name を音声化することを考える。この場合も、一つのディレクトリにすべての音声データを蓄積することも可能である。しかしこの場合においても前に述べたように、それが少ないうちは問題ないが、規模が拡大していくにつれて、問題が生じてくることは明らかである。そこで、今回考えた方法は、name をドメインごとに分類し、ドメインごとに音声デー

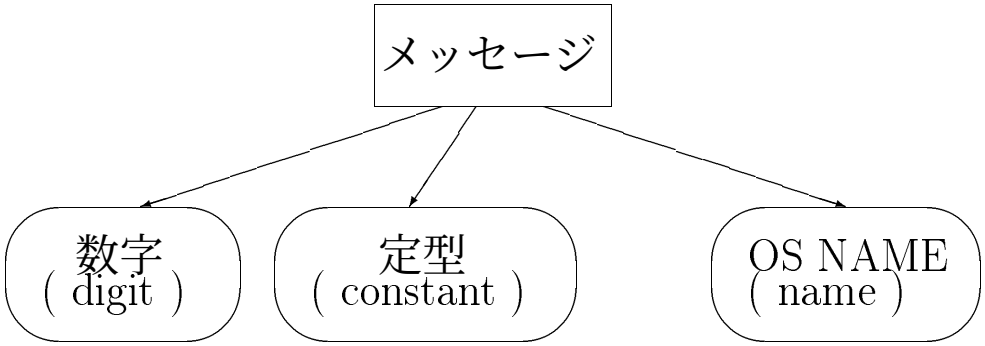


図 4.1: 音声化の方法

タベースを蓄積するという方法を考えた。具体的に、その分類の音声データベースの構造を図に示す（Figure4.2参照）。

以上のような構造の音声データベースをつくり phone メッセージの音声化をはかることにした。

4.4 音声化の方式

前の節で述べたような構造で、データベースを作成するということは、決まった。それでは実際の音声データをどのように作成（符合化）し、また音声合成の方式には何が適当かということについて考えていく。

4.4.1 音声合成の方式

音声合成には、録音編集方式、パラメータ編集方式、規則編集方式の三つがあるということは、前に見てきた通りである。さて今回の実験で行うのは、音声呼び出し機能付き phone（phone メッセージの音声化）である。メッセージの音声化するには、どの合成方法が適当であろうか。これに最も適した合成方法はどれであろうか。これについては、phone メッセージの音声化の方法のところで（間接的にではあるが）述べたように、録音編集方式（必要となる単語をあらかじめ録音しておき、それが必要な時はそれらをつなぎ合わせて音声化する、という方法）である。これを選んだのは次のような理由からである。

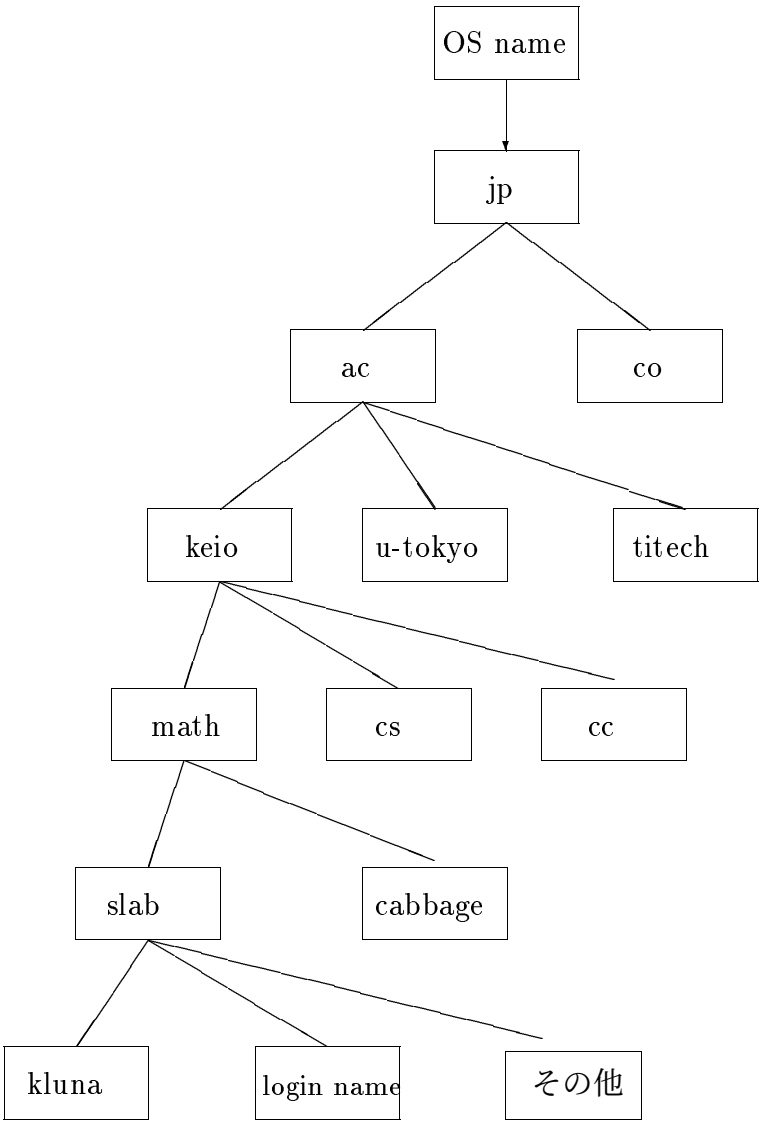


図 4.2: name の音声化の方式

- 録音編集方式は容易に現在の環境で実現でき、なおかつ高質な音が望める。一方、パラメータ編集方式、規則編集方式は高度の技術を必要とし、そのための機械も必要である。

  また録音編集方式でも、日本語の文章であれば、日本語特有のモーラ（５０音、濁音、半濁音、拗音など計約１００個）と呼ばれるものさえあれば理論的には任意の文章が合成可能な訳である。しかし、実際に行なってみると、いかにもロボットが喋っているといったものにしかならず、注意して聞いていないと何を喋っているかわからない低質の音にしかならなかった。

- 現在 phone メッセージは、英文字で構成されている（すべてのメッセージについていえることかもしれないが）。このため、それを規則合成方式によって音声合成するには、日本語に比べて基本単位が多数必要となる。

- 前の節で phone メッセージの構造を分析した結果、それはおよそ、文字、数字、OS name の三つのグループからなることがわかった。また、phone メッセージについては勿論、その他のメッセージを加えても、これを構成する単語の数はたかが知れている。また、最近のコンピュータのメモリの容量の大きさを考えると、録音編集方式であっても十分に通用するであろうと考えられる。

したがってこれらの理由から、この研究では録音編集方式を用いることが最も適当であると考えられる。

4.4.2 符合化の方式

次に、今回実験で用いた符合化の方式について述べる。今回用いた機材によって、取り扱い可能な符合化方式は表のとおりである（Table??参照）。

この中から適当なものを選ぶわけだが、これを選ぶにあたって考慮しておかなければならないことがある。それは、3.2.2 節の論理的構成要素のところで述べたことである。つまり、システム形態は最終的には水平分散型になることが望ましい。しかし、システム設計の初期の段階ではまずデータベースをある場所で集中的に管理させ、遠隔からこの中央データベースを利用するシステム形態を作ることが先決であるということである。従って、今回つくるデータベースは、今後しばらくの間、中央データベース（標準形式）として用いられるものとなる。

標準形式として用いるわけであるから、あまり質の低いものは避けるべきであろう。理由は簡単である。良い音を悪い音にするのは簡単だが、

表 4.1: 取扱い可能な音声データ

samplng freq.	37.8 kHz	18.9kHz	9.45kHz	8kHz
Straight PCM 16bit	◯	◯	◯	◯
Straight PCM 8bit	◯	◯	◯	◯
ADPCM 4bit	⋆	-	-	-
ADPCM 8bit	⋆	⋆	⋆	-
μ-Law 8bit	◯	◯	◯	◯
A-law 8bit	◯	◯	◯	◯

（◯ は、再生も録音も可能。⋆ は、録音のみ可能。）

逆は不可能だからである。 例えば、37.8 kHz でサンプリングしたものを 8 kHz サンプリングにするということなら、37.8 kHz のデータからある一定の間隔でデータを取ってっくるという簡単な作業で出来る。ところが、逆に 8 kHz でサンプリングしたデータを 37.8 kHz にするのは、不可能である。それは、失われたデータを、元に戻せといっているのと同じことだからである。

以上これらのことから、符合化方式は次のように決定した。

- フォーマット ··· μ Law
- サンプリング周波数 ··· 37.8 kHz
- チャンネル数 ··· モノラル
- データ幅 ··· 8 [bit]

4.5　ネットワーク環境への導入の方法

最後に、この音声呼び出し付き機能つき phone を、どのようにネットワーク環境に導入したかについて述べる。まず、その概略を述べる。三章で論じたことだが、ネットワークへの音声導入の方式としては、二つのモデルが考えられる。一つは個室モデルであり、もう一つは教室モデルである。もう一度これら二つのモデルのそれぞれの特徴を簡単に述べる。教室モデルというのは、音声をみんなで利用する形態である。一方、個室モデルの場合は特定の人のみが、その利益を教授するとい形態である。今回作った、音声呼出し機能付き phone の場合は、教室モデルの方に属する。すな

わち、phoneがかかってきた時には、「○○さんから、○○さんへphoneがかかってきている」ということが、全ての人に聞こえる状態にある。自分にphoneがかかってきた時以外は、あまり関係ない、と言えばそのとおりである。しかし一応すべての人が、このシステムの利益を教授できる状態にはなっているわけである。

さて、概略としては以上のようなものである。それでは実際にはどのような環境が出来上がったのか、ということについて次に述べる。まず機材はどうなっているか。この点に関してはあまり恵まれていず、今の環境には、音声を扱える機材は部屋に一台しかない。今、便宜上この音声を扱える機材のことをdixonと呼ぶことにする。この時、dixonにphoneをかければ音声で呼び出してくれるというシステムを作ることは容易である。しかしこれでは、dixonに対してphoneをかけたときだけ、この機能が働くことになる。これでは、「条件付き音声呼出し機能付きphone」という長ったらしい名前が必要になってしまう。すべてのマシンが音声を取り扱えるならそれでもいい。しかし現段階では、前にも述べたように、音声を扱うことができる機材は、身近には一台しかない。これでは、未完性の教室モデルである言わざるをえない。そこで、dixonには音声を出させるだけ、というシステムを作ることを考える。つまり、dixon以外のマシンにphoneをかけたとしても、それがdixonの置いてある部屋にあるものならば、dixonが音声でそれを教えてくれるというものである。(このとき、dixonは音声を出すだけである。)このようなシステムにすれば十分に音声呼出し機能付きphoneの役割を、教室モデルでも果たすようになるのである。

第 5 章

今後の課題について

今回の研究では、マルチメディアのネットワーク環境への導入の一形態として音声呼び出し機能つき phone（phone メッセージの音声化）を実現するための研究を行なった。それでは、今後これをどう生かし、更に研究を進めていくべきであろうか。

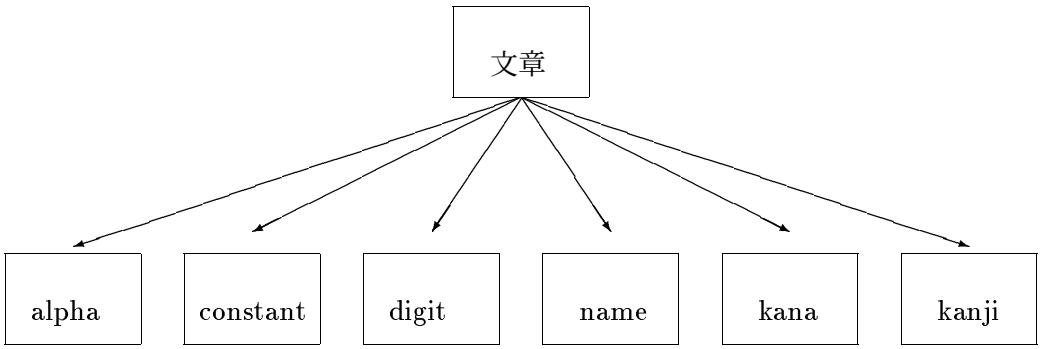
5.1 音声合成の方式について

今回の研究で用いた音声合成の方式は、録音編集方式である。録音編集方式の特徴は、容易に実現可能であり、かつ高質な音が再生可能ということである。しかし一方で、必要な単語のすべてを録音するという方式のため他の二方式に比べて多量のメモリを必要とするという欠点を持っている。このことは、前に述べた通りである。従って、他の二方式の一日も早い実用化が望まれるのである。しかし、ここしばらくは、録音編集方式で行なうほかはない。また、音声データの圧縮方式の研究も日々続けられており、今後の発展が期待されるところである。これら二つの問題は、音声そのものの研究に近く、ここではあまり深いりする必要はないであろう。とはいうものの、これらは、音声のネットワーク導入という意義からも、大きな意味を持っており、全く無視してはいられないことも事実である。今後のどのようになっていくのか期待される。

そういうわけで、現段階では、今の録音編集方式を用いて、消費メモリをなるべく少なく、しかもいい音質で再現できるような方法を開発することが必要となことである。

5.2 音声化の方法について

今回用いた、音声化の方法は、次のようなものである。まず音声データは、定型、数字、OS name の三つのグループに分類し音声データをつくる。また OS name のうち、特に login name に注目し、それらをドメインごと



に分類し、同じようにそれごとに音声データをつくる。(この場合 tree 構造をつくることになる) こうして貯めておいたデータを、音声化する際にそれぞれのグループから検索し、それらを組合わせて一つのメッセージとして音声化するという方法である。この方法の特徴というのは、データを tree 構造にすることによって、システムの拡大が容易にでき、しかもデータを整理するのが非常に楽であるということである。例えば、あるユーザの login ネームの音声データが欲しい時には、この tree をたどっていけば容易にそれを見つけることが可能である。また、それを新たに付け加えことも容易に行なえる。また同じように、今のシステムに新たに仮名用の音声データを加えたければ上の三つのグループにそれをちょっとつけ加えてやるだけで良いのである。(上図参照)

また、もしそれを行なった後に、それよりも音質の良い仮名用の音声データベースが見つかったとしたら、そのデータベースと今使っている音声データベースとを、そっくり取り替えてやるだけで良いのである。この時、仮名以外のデータには、何の変更も加える必要がない。それぞれ独立したものとして取扱えるのである。こうすることによって、変更が容易に可能なわけである。また、このようにグループとして音声データを取り扱うことにより、録音編集方式の弱点を幾分でも少なくできる。つまりその弱点と言うのは、録音編集方式の場合人間の声をそのまま用いるため、できるならすべての音声データが、同じ人から発せられた音声であることが望ましい (いろいろな人の声が混じってしまうと聞き取りにくいものになってしまうから)。しかし、システムが大きくなるにつれてそれは困難なことになってくるであろう。そこで、少しでもこのようなことを防ぐようの

することが望まれる。もし今回用いたような、データをグループごとに取り扱えるシステムであれば、録音編集方式特有のこの弱点を少しでも減らすことができる。

このように、今回作ったシステムは容易に拡張（変更）が可能なものである。今回の研究では、データをグループごとに分類して取り扱った。またユーザの login ネームは、これをドメインごとに分け tree 構造にして整理したわけである。このような考え方の基礎を作ったという点で、この研究は十分意義があるものと考える。これから、この tree をどのように成長させていくかということは全てユーザ（管理者）の自由である。

5.3　システム拡大時の問題

上で述べたように、tree 構造の導入によってシステムを管理すること（拡張したり変更したりすることを含めて）は容易に可能になった。しかし、それはデータベースの作成という点からであり、実際にはそれ以外にも様々な問題がある。そのなかで、研究の中で思いついたものを取り挙げてみる。

今回作ったシステムは、英文字ようの音声データしか取り扱わない。では、これに日本語や、ローマ字をも利用可能な形態にしたらどうなるであろうか。データをつけ加えることは簡単である。（前に述べたように）しかし、それ以外にも問題がいくつか考えられる。例えば、次のような例を考えてみよう。「take」という文字があったとする。これは何と読めばいいのだろうか。「たけ」とも読めるし、「テイク」とも読める。また「ティー エー ケー イー」とも読める。読み方は、それが使われている場所や、それを書いた人によって異なってくるであろう。したがってこれをコンピュータに正しく読ませるには、自動的にそれを判断させるか、あるいはそれを使うユーザにある程度の制約を課すしかないであろう。しかし、コンピュータに自動的に判断させるといってもそれにも限界がある。従って、システム設計者がある程度規則を定めておき、それを使う側に強制させることが初期の段階では必要であろう。

また、数字の音声化も厄介な問題である。なぜなら、数字全てを音声データとして録音することは不可能だからである。この問題は、音声合成の開発が進めば解決できるであろうが、前にも述べた通り、その実現にはまだまだ時間がかかりそうである。従って、現段階でも実現可能な録音編集方式で、解決する方法を考えるしかない。この場合、その全てを録音するというのはやはり無理がある。従ってやるべきことは、なるべく良い音質で、かつなるべくメモリーが少なくて済むような方法を見つけることで

ある。この場合、音声の持つ性質を利用して、必要となる音質データを減らす方法が考えられる。「101]「102」「103」.... などは、「ひゃく」という音声と、「いち」「に」「さん」... という音声を組み合わせることによって実現可能である。しかし、この時注意しなければならない問題がある。それは同じ「ひゃく」でも、「100」を読む時の「ひゃく」と、「101」を読む時の「ひゃく」では、微妙にイントネーションが異なるということである。したがって、文字にすれば同じであるが、音声データとしては二種類用意しておかなければならないのである。

このように、音声データであるがゆえに起こってくる問題がある。従って、音声を取入れようとする時は、これらの問題にも目を向けていくことを忘れてはならない。

5.4 ネットワークへの導入の問題

それでは、ネットワーク環境へ音声データを導入する時に起こる問題について考えてみる。この時に起こり得る問題は大きく分けて、二つのものが考えられる。まず、通信時に起こる問題。それから、それを、送った後に起こり得る問題である。ここでは、データを目的地に送った後に起こる問題について先に考えてみる。

その問題というのは、一言で言えば、データの標準形式の問題と言うことができる。音声データの標準形式と言うものは、存在しない。今回の研究で作成したデータは、音声データの絶対的な標準形式と言うわけではない。ただ単に、標準形式と考えて用いてみよう、というだけのいわば相対的な意味のものである。従って、このデータが必ずしも、全ての機材に適用できるというわけではない。標準形式とは異なった音声データを必要と場合には、このデータをそれを音声化するための準備として、それぞれの voice device にマッチしたデータに変換してやるという作業をまずしてやらなければならない。このシステムを作ることは、今後大きな課題となることだろう（Figure5.1参照）。

例を挙げれば、こういうことである。今回作成したデータは、37.8kHz のサンプリングで行なった。しかし、中には 16kHz サンプリングのデータしか使用不可能なものもあるし、8kHz サンプリングのデータしか使用不可能なものもある。このような時には、37.8kHz サンプリングのデータを、それに合うように手を加えてやらなければ、その音声データを用いることはできないのである。

次に、データ通信上の問題としては、トラフィックの輻輳の問題があげられる。この問題の解決方法として考えられるのは、より効率的な分散型

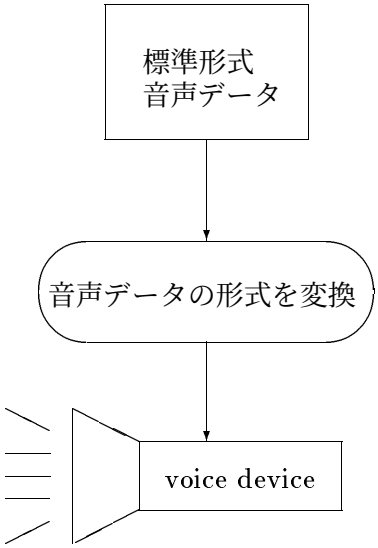


図 5.1: 標準形式の変換

データベースの開発、またその他のものとして、ISDN の開発、新たな音声合成技術の実現などがあげられる。また、上で述べたように目的地に到着してから、37.8kHz サンプリングのものを 16kHz や 8kHz に変えるのではなく変換してから送れば、多少はトラフィックの問題が解消できるかもしれない。

第 6 章

まとめ

音声は現在、様々な場所で利用されるつつある。そこで今回の実験では特に、ネットワーク環境に音声を導入してみた。音声は一般のデータに比べて多量のメモリを必要とする。また、音声特有の性質もある。そこで、どうすればより効率的に、音声を利用することができるか。また、それを行なうにあたっての問題点は何か。これらのことを考えるために、この研究を行なってみた。

今回の研究では、音声呼び出しつき phone（phone メッセージの音声化）の設計を行った。音声合成の方式は、録音編集方式である。これを選んだ理由は前に述べた通りである。また、今回行なった音声化の方法は、次のような方法である。まず音声データは、定型、数字、OS name の三つのグループに分類して tree 構造を作って蓄積しておく。また OS name のうち特に login name に注目し、それらはドメインごとに分類し、そして上と同じように tree 構造を作って整理しておく。このようにして貯めておいたデータを音声化する際にそれぞれのグループから検索し、それらを組合わせて一つのメッセージとして音声化するという方法である。ここでのポイントは、データを tree 構造にして整理するということである。このような方法を用いることによりデータの変更や、検索を容易に行なうことができるようになった。今後この研究をもとに、更に研究を重ね、どんなものでも音声化できるようになることを望む。

また、この研究は音声化のためだけの研究ではない。これは、来るべきマルチメディアネットワーク時代のための研究ともいえるのである。そういう意味でも、この研究が大いに役立つことを望む。

なお、この研究をするに際しては、音声合成の技術や、音声の持つ特有の性質などにも、目を向けていくべきである。これらのことは、この研究を進めていく上で重要な意味をもつものであるから。